Application Note

XGate-DPS 嵌入式模块

PROFIBUS-DP 从站通讯模块

V1.00 Date: 2009/09/01 产品应用笔记

类别	内容
关键词	UART XGate-DPS
摘 要	本文档帮助用户掌握如何通过 XGate-DPS 系列模块的串行接口接入 PROFIBUS 总线

广州致远电子有限公司

修订历史

版本	日期	原因
V1.00	2009/09/01	创建文档
V1.01	2009/12/01	规范代码、增加 GSD 修改和组网

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

销售与服务网络（一）

广州周立功单片机发展有限公司

地址：广州市天河北路 689 号光大银行大厦 12 楼 F4
邮编：510630
电话：(020)38730916　38730917　38730972　38730976　38730977
传真：(020)38730925
网址：www.zlgmcu.com

广州专卖店

地址：广州市天河区新赛格电子城 203-204 室
电话：(020)87578634　87569917
传真：(020)87578842

北京周立功

地址：北京市海淀区知春路 113 号银网中心 A 座 1207-1208 室 （中发电子市场斜对面）
电话：(010)62536178　62536179　82628073
传真：(010)82614433

杭州周立功

地址：杭州市天目山路 217 号江南电子大厦 502 室
电话：(0571) 28139611　28139612　28139613
28139615　28139616　28139618
传真：(0571) 28139621

深圳周立功

地址：深圳市深南中路 2070 号电子科技大厦 C 座 4 楼 D 室
电话：(0755)83781788（5 线）
传真：(0755)83793285

上海周立功

地址：上海市北京东路 668 号科技京城东座 7E 室
电话：(021)53083452　53083453　53083496
传真：(021)53083491

南京周立功

地址：南京市珠江路 280 号珠江大厦 2006 室
电话：(025)83613221　83613271　83603500
传真：(025)83613271

重庆周立功

地址：重庆市石桥铺科园一路二号大西洋国际大厦（赛格电子市场）1611 室
电话：(023)68796438　68796439
传真：(023)68796439

成都周立功

地址：成都市一环路南二段 1 号数码同人港 401 室（磨子桥立交西北角）
电话：(028)85439836　85437446
传真：(028)85437896

武汉周立功

地址：武汉市洪山区广埠屯珞瑜路 158 号 12128 室（华中电脑数码市场）
电话：(027)87168497　87168297　87168397
传真：(027)87163755

西安办事处

地址：西安市长安北路 54 号太平洋大厦 1201 室
电话：(029)87881296　83063000　87881295
传真：(029)87880865

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

销售与服务网络（二）

广州致远电子有限公司

地址：广州市天河区车陂路黄洲工业区 3 栋 2 楼
邮编：510660
传真：(020)38601859
网址：www.embedtools.com （嵌入式系统事业部）
www.embedcontrol.com （工控网络事业部）
www.ecardsys.com （楼宇自动化事业部）

技术支持：

CAN-bus：
电话：(020)22644381 22644382 22644253
邮箱：can.support@embedcontrol.com

iCAN 及数据采集：
电话：(020)28872344 22644373
邮箱：ican@embedcontrol.com

MiniARM：
电话：(020)28872684 28267813
邮箱：miniarm.support@embedtools.com

以太网：
电话：(020)22644380 22644385
邮箱：ethernet.support@embedcontrol.com

无线通讯：
电话：(020) 22644386
邮箱：wireless@embedcontrol.com

串行通讯：
电话：(020)28267800 22644385
邮箱：serial@embedcontrol.com

编程器：
电话：(020)22644371
邮箱：programmer@embedtools.com

分析仪器：
电话：(020)22644375 28872624 28872345
邮箱：tools@embedtools.com

ARM 嵌入式系统：
电话：(020)28872347 28872377 22644383 22644384
邮箱：arm.support@zlgmcu.com

楼宇自动化：
电话：(020)22644376 22644389 28267806
邮箱：mjs.support@ecardsys.com
mifare.support@zlgmcu.com

销售：

电话：(020)22644249 22644399 22644372 22644261 28872524
28872342 28872349 28872569 28872573 38601786

维修：

电话：(020)22644245

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

目 录

1. 适用范围.....1
2. 原理概述.....2
3. 开发环境.....3
3.1 硬件开发环境.....3
3.2 软件开发环境.....3
4. 使用DEMO开发产品.....4
4.1 DEMO模拟资源.....4
4.2 UART接口通信流程.....5
4.3 DEMO开发包.....6
4.3.1 DEMO代码包含文件列表.....6
4.3.2 主要功能函数列表.....6
4.3.3 代码使用说明.....7
4.4 修改GSD.....18
5. 组态网络.....24
5.1 主站平台.....24
5.2 组态系统.....24
6. 免责声明.....33

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

1. 适用范围

本文档仅在指导用户如何使用广州致远电子有限公司的嵌入式 XGate-DPS 模块。用户参照该文档，仅需修改少量代码就可以方便快捷的将嵌入式 XGate-DPS 系列模块嵌入到自己的产品中，设计预想的设备。

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

2. 原理概述

用户使用 UART 接口时，与模块的通信是半双工的主从方式，用户主机主动命令、请求模块，模块被动回答用户主机的命令或请求，这一点相当重要。当模块有接收到 PROFIBUS 主站数据时，并不能主动将数据发送给用户主机，而需要用户实时发送命令来请求这也即是说，您设计的系统的实时性取决于你请求模块的快慢。

这里举例说明，用户何时使用 UART 接口是合理的。假如您需要将您的数据采集模块接入 PROFIBUS 总线，数据采集从站一共有 N 个字节的采集数据需要上传，采集更新时间为 T， UART 最小通讯速率 V_{min} ，保留 30%的余量，则有：

$T = （ N + 5 ） \times 8 \times (1/V_{min}) \times 130\%$

假如设定 UART 的波特率一般在 1Mbps，我们的从站需上传 8 个字节的数据，得出运行最大的采集更新时间 T_{max} ：

$T_{min} = 135.2$ uS 。

当要求采集更新时间 T < 135.2 uS 时，那么就满足不了需要。

随你购买的模块套件，提供了使用 DEMO，这些 DEMO 函数有效的实现了常用操作功能，你只需按照你的需要修改少量代码，就可以直接运用到你的系统中。本文档将通过基于配套的评估板 DEMO 来说明使用方法

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

3. 开发环境

3.1 硬件开发环境

XGate-DPS 系列模块支持所有能提供模块相应接口的 MCU，包括 ARM 系列、51 系列等。本文档以模块配套的评估板为平台来介绍使用方法，该评估板使用的是 NXP 的 51 系列单片机(详细评估板原理图见光盘内容)。评估板如：

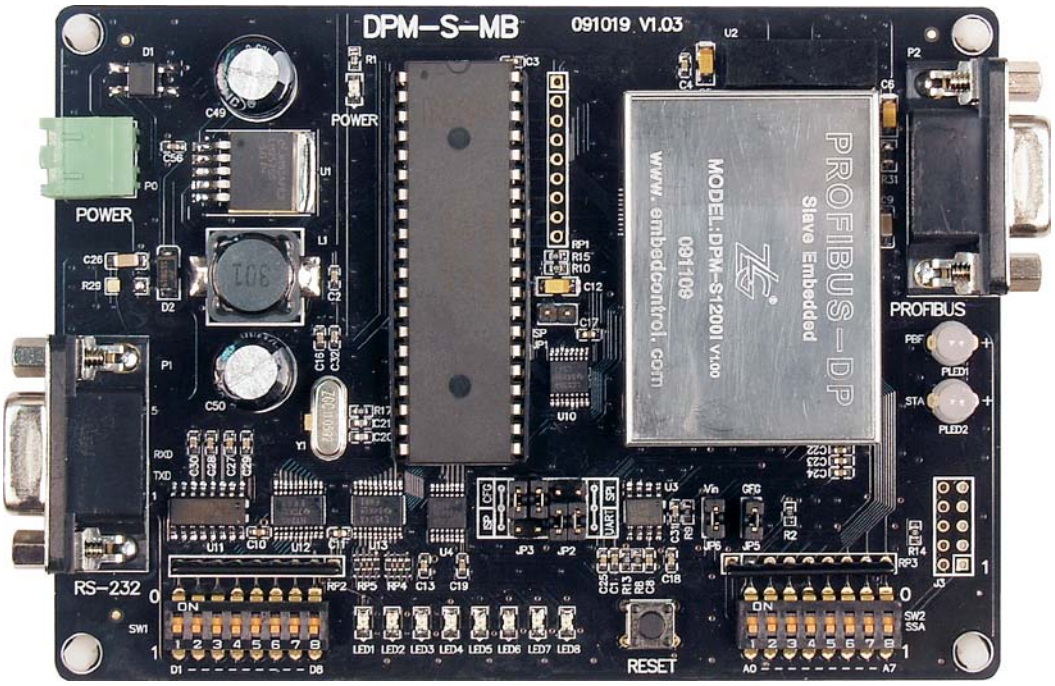

图 3.1 评估板实物图

用户使用 UART 接口时，需要对评估板做如下跳线：

1. 将 JP3 短接帽悬空，不要有任何方向短接 ；
2. 将 JP2 短接帽短接到“UART”端；
3. 短接 JP6 Vin，为模块接通电源。

注意，不要短接 JP5 CFG。

特别说明的是，用户开发产品时，必须提供一路隔离 5V 电源，推荐使用致远电子有限公司的隔离 DC-DC 模块“ZY0505FKS-1W”。其接口电路图详见光盘的评估板原理图。

3.2 软件开发环境

用户可以使用 KEIL、ADS 等集成开发环境。本评估板示例 DEMO 是基于 KEIL 开发平台的 C 代码。

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

4. 使用DEMO开发产品

4.1 DEMO模拟资源

配套的开发板可 I/O 资源如下：

表 4.1 I/O 资源

序号	资源	描述
1	8 位开关量输出	开发板 LED 灯组显示
2	8 位开关量输入	开发板拨码开关输入
3	16 位计数器输入	开发板 MCU 主循环计数产生，具有上限和下限

表 4.2 DP-V1 资源分配

Slot	Index	资源	DS_READ	DS_WRITE
1	2	8 位开关量输出	√	√
2	2	8 位开关量输入	√	
3	2	16 位计数器输入	√	
4	1	16 位计数器下限设置	√	√
4	2	16 位计数器上限设置	√	√
5	1	从站地址开关	√	

16 位计数器上下限设置数据结构：

表 4.3 16 位计数器上下限设置

字节	意义
0	时基，1：10m；0：1ms
1,2	下限值
3,4	上限值

表 4.4 DEMO 用户参数

序号	资源	用户参数	参数描述
0	DP-V1 状态	字节 0-2	
1	8 位开关量输出	字节 3	输出有效逻辑选择
2	8 位开关量输入	字节 4	输入有效逻辑选择
3	16 位计数器输入	字节 5	计数器使能：1 使能，0 禁能
		字节 6	计数基值选择
		字节 7，8	下限值
		字节 9，10	上限值

表 4.5 诊断与报警设置

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

序号	资源	描述	数据格式
1	16 位计数器到达上限或下限	报警通知主站超限	字节 0-3：标准报警头； 字节 4：当前时基，单位 ms； 字节 5，6：报警时，计数器值； 字节 7，8：报警时下限值； 字节 9，10：报警时上限值； 字节 11 开始：与通道相关的诊断(用输入模块来模拟断路的诊断信息)
2	16 位计数器超限消失	消除报警	字节 0-3：标准报警头； 字节 4：当前时基，单位 ms； 字节 5，6：当前正常值
3	地址设置开关拨为非预定地址	诊断状态信息，用户动用了地址设置	字节 0-3：标准状态报告头； 字节 4：当前地址被拨动到的值
4	地址设置开关还原	状态信息消除	字节 0-3：标准状态报告头； 字节 4：当前地址被拨动到的值

4.2 UART接口通信流程

用户使用UART接口与XGATE-DPS系列模块的操作步骤如图 4.1所示：

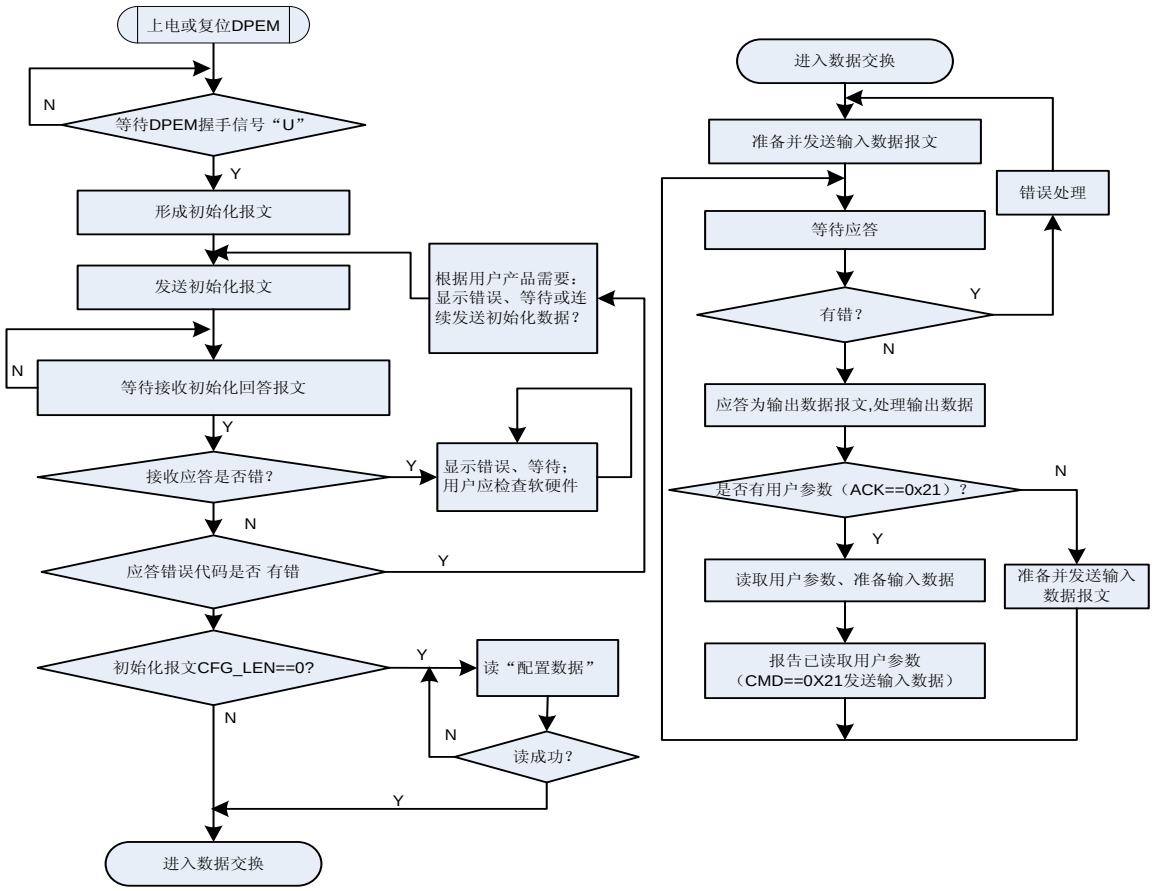

图 4.1 串行接口下用户对模块的操作过程

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

注：1) 形成初始化报文时，如果用户要求自适应配置数据，则必须设置 CFG_LEN 为 0；这种情况适合配置数据有主机决定(如网关类产品)和配置数据个数大于 24 的场合。

2) 当 CFG_LEN 为 0，用户在进入数据交换前必须先读“配置数据”，详见《XGATE-DPS 系列模块数据手册》的命令“0x23”。

3) 用户可以通过读到的“配置数据”来计算实际“总 I/O 长度”，即初始化报文中的“D_out_len”和“D_in_len”。也可以使用命令“0x24”读“I/O 数据总长度”，详见《XGATE-DPS 系列模块数据手册》的命令“0x24”。

4）请用户详细阅读《XGATE-DPS 系列模块数据手册》中的命令“0x20”～“0x22”。

4.3 DEMO开发包

4.3.1 DEMO代码包含文件列表

表 4.6 DEMO 代码文件列表

序号	文件名	实现功能
1	UART.C	实现 UART 通信的功能函数驱动库
2	UART.H	UART 结构头文件
5	CRC.C	CRC 校验函数
3	HEAD.H	评估板示例头文件
4	MAIN.C	评估板功能函数及主循环函数
6	User.c	用户在该文件中修改或添加各功能函数的处理
7	Cfg.h	用户通过修改该文件的宏定义来设置设备必要参数

4.3.2 主要功能函数列表

表 4.7 DEMO 代码主要函数列表

序号	函数名	所属文件	实现功能
1	PBIniti()	UART.C	该函数完成模块初始化
2	ReadeCFG()	UART.C	用户调用该函数获取来自 PROFIBUS 主站的配置数据和输入输出长度
3	Serial_Send ()	UART.C	该函数负责将要发送的数据按 DPM 规定帧格式打包并调用 UART 发送中断发生数据
4	DATAProcess()	UART.C	该函数处理接收到的数据。它负责将各类数据放到指定的存储区
5	ISR_UART ()	UART.C	负责接收和发送 UART 通信数据的中断服务程序
6	DataUpdate（）	UART.C	用户调用该函数实现输入数据上传
7	dpv1_statemachine（）	UART.C	该函数完成 DPV1 的处理
8	Alarm_Stat_Send（）	UART.C	该函数完成对诊断/报警的处理，用户有诊断/报警时调用该函数

表 4.8 固件回调函数

序号	函数名	所属文件	实现功能
1	User_DPV1_read_req（）	User.C	用户在该函数中实现对非周期读请求的处理
2	User_DPV1_write_req（）	User.C	用户在该函数中实现对非周期写请求的处理
3	User_DPV1_data_transport_req（）	User.C	用户在该函数中实现对非周期数据传输请求的处理

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

表 4.9 用户仿写函数

序号	函数名	所属文件	实现功能
1	main（）	main.C	用户需要仿照 DEMO 写该主函数
2	user_demo_diagnostic（）	User.C	用户需要仿照该函数写自己的诊断报警函数

4.3.3 代码使用说明

本 DEMO 中，已经建立了一个全局结构体变量，用户需要的输出数据、用户参数、配置数据都在该结构体变量的成员里，用户需要发送的输入数据、发件箱等，也都应该放到该变量的相应成员中。

程序 4.1 STRUCT_SYS 结构体

```
typedef struct
{
    unsigned char State       ;                    /*  用户程序状态机。                */
    unsigned char Flag        ;
    CMD_TYPE          CMD;
    ACK_TYPE          ACK ;
    unsigned char DPV1_State ;
    unsigned char Mail_Len;
    unsigned char AL_ACK ;                         /*  通过查询该标志来判断主站是否响应了报警*/
    unsigned char In_data[SIZE_INPUT_DATA];        /*  循环输入数据                    */
    unsigned char Out_data[SIZE_OUTPUT_DATA];      /*  循环输出数据                    */
    unsigned char User_para[SIZE_USER_PARAM];      /*  来自主站的用户参数数据          */
    unsigned char Mail_send[SIZE_MAIL_SEND];       /*  发送邮件箱                      */
    unsigned char Mail_rec[SIZE_MAIL_REC];         /*  接收邮件箱                      */
    STRUCT_DEVICE_CHARACTER Device ;               /*  设备属性                        */
} STRUCT_SYS;
```

程序 4.2 STRUCT_DEVICE_CHARACTER

```
typedef struct
{
    unsigned char   Address                 ;      /* ID,从站地址                      */
    unsigned char   ProductID[2]            ;      /* Product ID                       */
    unsigned char   IO_CFG_LEN              ;      /* I/O 配置数据长 CFG_LEN           */
    unsigned char   SD_In_Len               ;      /*  输入数据长 SD_in_len            */
    unsigned char   SD_Out_Len              ;      /*  输出数据长 SD_out_len           */
    unsigned char   User_Prm_Data_Len ;            /*  用户参数长度 User_Prm_Data_Len */
    unsigned char   DPV1_STATUS    ;               /* DPV1 参数状态                    */
    unsigned char   Type_Enable_Alarm    ;         /*  允许的报警类型,                */
    unsigned char   Num_Enable_Alarm;
    unsigned char   Feature_supported    ;         /* default: 0x01, Read and Write supported */
    unsigned char   Feature_supported1 ;           /* default: 0x00                    */
    unsigned char   Profile_feature_supported ;    /* default: 0x00, vendor or profile specific */
    unsigned char   Profile_feature_supported1 ;   /* default: 0x00, vendor or profile specific */
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
    unsigned char    Profile_Ident_Number_H   ;          /* default: 0x0000, vendor or profile specific   */
    unsigned char    Profile_Ident_Number_L ;            /* default: 0x0000, vendor or profile specific   */
    unsigned short   C2_Response_Timeout;                /*  指定从站 MS2 服务开始初始化到       */
                                                         /*  得到数据的最大时间                  */
    unsigned char    C1_DATA_LEN;                        /* Length of MSAC_C2 Data;Function_Num，*/
                                                         /* Slot_number, Index, Length excluded  */
    unsigned char    C2_DATA_LEN;                        /* Length of MSAC_C1 Data;Function_Num, */
                                                         /* Slot_number, Index, Length excluded  */
    unsigned char    Cfg_Data[SIZE_CFG_DATA];            /*  配置数据                           */
} STRUCT_DEVICE_CHARACTER;
```

STRUCT_SYS 的内容和定义，用户不可以修改。

用户要初始化模块，只需在 cfg.h 文件中修改相应的宏定义。

程序 4.3　在 cfg.h 文件中宏定义

```
/*****************************************************************************
* 1    include hierarchy
*****************************************************************************/

/*****************************************************************************
* 2    编译器选择
*****************************************************************************/
#define C_51                                             /*  如果用户使用的是非 C_51 编译器  */
                                                         /*  ，请注释该语句                  */
#ifndef C_51
#define C_ARM
#endif
/*****************************************************************************
* 3    系统时钟和使用 UART 的通信速率
*****************************************************************************/
#define UART_RATE       ((ULONG)19200)
#define sysclk          ((ULONG)11059200)
/*****************************************************************************
* 3    修改与硬件相关的寄存器
*****************************************************************************/
#define UARTRI              (RI )                        /*  用户 MCU 的 UART 接收中断标志       */
#define CLRUARTRI()         (RI=0)                       /*  清除用户 MCU 的 UART 接收中断标志  */
#define UARTTI              (TI )                        /*  用户 MCU 的 UART 发送中断标志       */
#define CLRUARTTI()         (TI=0)                       /*  清除用户 MCU 的 UART 发送中断标志  */
#define UARTRBUF            (SBUF)                       /*  用户 MCU 的 UART 接收 buffer        */
#define UARTTBUF            (SBUF)                       /*  用户 MCU 的 UART 发送 buffer        */
#define Enable_UART_int(set) (EA=set)                    /*  用户 MCU 的 UART 中断使能           */
/*****************************************************************************
* 4    定义各缓冲器大小
*****************************************************************************/
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
#define SIZE_INPUT_DATA    ((UBYTE)32)                    /* 输入数据长度                    */
#define SIZE_OUTPUT_DATA ((UBYTE)32)                      /* 输出数据长度                    */
#define SIZE_USER_PARAM    ((UBYTE)20)                    /* 参数数据长度                    */
#define SIZE_CFG_DATA      ((UBYTE)20)                    /* 组态数据长度                    */
#define SIZE_MAIL_SEND     ((UBYTE)20)                    /* 发件箱长度                      */
#define SIZE_MAIL_REC      ((UBYTE)50)                    /* 收件箱长度                      */
                                                          /* 定义 UART 接收和发送缓冲区大小  */
#define REC_MAX_LEN        ((UBYTE)122)                   /* 定义接收数据缓冲长度，其长度    */
                                                          /* = 最大数据单元长度 +5（帧       */
                                                          /* 头帧尾）                        */
#define SEND_MAX_LEN       ((UBYTE)122)                   /* 定义发送数据缓冲长度，其长度    */
                                                          /* = 最大数据单元长度 +5（帧       */
                                                          /* 头帧尾）                        */
/*****************************************************************************
* 5  设置您的设备属性
*****************************************************************************/
#define Device_Address                ((UBYTE)rd_address()&0x7f) /* 从站地址              */
#define Device_ProductID              ((UWORD)0x0f01)           /* ID                    */
#define Device_SD_In_Len              ((UBYTE)0x03)             /* 输入数据长            */
#define Device_SD_Out_Len             ((UBYTE)0x01)             /* 输出数据长            */
#define Device_User_Prm_Data_Len      ((UBYTE)0x0b)             /* 用户参数长度          */
/*
 * DP-V1 参数
 */
#define Device_DPV1_STATUS        ((UBYTE)0|ENABLE_Fail_Safe) /* 检查组态模式，严格一致      */
#define Device_Type_Enable_Alarm ((UBYTE)(PROCESS_ALARM_Enable|DIAGNOSTIC_ALARM_Enable))
                                                          /* 允许的报警类型,             */
#define Device_Num_Enable_Alarm    ((UBYTE)(0))           /* 允许 2^Device_Num_Enable_   */
                                                          /* Alarm 个报警，0 允许 1 个报警，1 */
                                                          /* 允许 2 个报警               */
#define Device_Feature_supported      ((UWORD)0x0001)     /* default: 0x01, Read and Write supported */
#define Device_Profile_feature_supported    ((UWORD)0x0000) /* default: 0x00, vendor or profile specific */
#define Device_Profile_Ident_Number         ((UWORD)0x0000) /* default: 0x0000, vendor or profile specific*/
#define Device_C2_Response_Timeout      ((UWORD)0x0064) /* 指定从站 MS2 服务开始初始化到   */
                                                        /* 得到数据的最大时间          */
#define Device_C1_DATA_LEN              ((UBYTE)0x40)   /* Length of MSAC_C1 Data;Function_Num, */
                                                        /* Slot_number, Index, Length excluded */
#define Device_C2_DATA_LEN              ((UBYTE)0x40)   /* Length of MSAC_C2 Data;Function_Num, */
                                                        /*Slot_number, Index, Length excluded */
```

上述宏定义必须和 GSD 描述一致。设定个长度值必须注意，REC_MAX_LEN 和 SEND_MAX_LEN 必须是其它任何缓冲区长度最大值+5；C1_DATA_LEN 和 C2_DATA_LEN 最大值为 0xF0，但必须小于 REC_MAX_LEN-5 和 SEND_MAX_LEN-5 ，并且要小于 SIZE_MAIL_REC。

在设定 cfg.h 文件后，用户需要修改 user.C 文件中的回调函数，这些函数主要是非周期

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

循环读写等的回调，用户需要在这些函数中参照 DEMO 修改成自己实际数据。

DP-V1 的非周期读/写/数据交换请求的应答基本一致，这里仅以读请求应答为例说明用户如何修改代码。当然，用户在设计自己的代码前，必须明确自己设计的从站有哪些 slot 和 index，各自被定义的含义是什么。

```
STRUC_MAIL* User_DPV1_read_req(STRUC_MAIL *preq_mail )
{
    STRUC_MAIL *pMail;
    UNION_MS12PDU *pMSPdu ;
    UNION_MS12PDU *preq ;
    /*
     * 该部分不必更改
     */
    pMail = (STRUC_MAIL *)sys.Mail_send;
    pMSPdu = (UNION_MS12PDU *)pMail->Data;
    pMail->ID = preq_mail->ID ;
    preq = (UNION_MS12PDU *)preq_mail->Data;
    /*
     * 参照应答格式写回复数据，这里只是个例子，用户根据自己实际评估并写数据
     */
    pMail->Len = 4 ;
    memcpy(&pMSPdu->ms_wr_rd_tr,&preq->ms_wr_rd_tr,4);          /* 复制头                                    */
    /*
     * 评估请求参数并应答请求
     */
    switch(preq->ms_wr_rd_tr.slot_num)
    {               /*
                     * 请求读 DO8
                     */
    case 1:
    if(preq->ms_wr_rd_tr.index == 2){
        memcpy(pMSPdu->ms_wr_rd_tr.pdu_data,sys.Out_data,1);
        pMSPdu->ms_wr_rd_tr.length = 1 ;                        /* 必须写实际可能提供的读数据长 */
                                                                /* 度                            */
        pMail->Len += 1 ;
    } else {                                                    /* 非法 index                    */
        pMSPdu->ms_neg_res.function_num = 0xDE ;                /* 指示错误发生                  */
        pMSPdu->ms_neg_res.err_decode = 0x80 ;                  /* Error_Decode ,指明是 PROFIBUS */
                                                                /* DP/DPV1 类错误                */
        pMSPdu->ms_neg_res.err_code1 = DPV1_ERRCL_ACCESS | DPV1_ERRCL_ACC_INV_INDEX ;
                                                                 /* Error_code1 ,支持是接入错，  */
                                                                 /* 接入了非法的 SLOT ；         */
        pMSPdu->ms_neg_res.err_code2 = 0x00 ;                    /* Error_code2 ,                */
    }
    break;
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
            /*
             * 请求读 DI8
             */
case 2 :
if(preq->ms_wr_rd_tr.index == 2){
    memcpy(pMSPdu->ms_wr_rd_tr.pdu_data,sys.In_data,1);
    pMSPdu->ms_wr_rd_tr.length = 1 ;                    /* 必须写实际可能提供的读数据长 */
                                                        /* 度                          */
    pMail->Len += 1 ;
} else {                                                /* 非法 index                  */
    pMSPdu->ms_neg_res.function_num = 0xDE ;            /* 指示错误发生                */
    pMSPdu->ms_neg_res.err_decode = 0x80 ;              /* Error_Decode ,指明是 PROFIBUS */
                                                        /* DP/DPV1 类错误              */
    pMSPdu->ms_neg_res.err_code1 = DPV1_ERRCL_ACCESS | DPV1_ERRCL_ACC_INV_INDEX ;
                                                         /* Error_code1 ,支持是接入错，  */
                                                         /* 接入了非法的 SLOT ；          */
    pMSPdu->ms_neg_res.err_code2 = 0x00 ;                /* Error_code2 ,               */
}
break;
            /*
             * 请求读当前 16 位计数器值
             */
case 3:
if(preq->ms_wr_rd_tr.index == 2){
    memcpy(pMSPdu->ms_wr_rd_tr.pdu_data,&sys.In_data[1],2);
    pMSPdu->ms_wr_rd_tr.length = 2 ;                    /* 必须写实际可能提供的读数据长 */
                                                        /* 度                          */
    pMail->Len += 2 ;
} else {                                                /* 非法 index                  */
    pMSPdu->ms_neg_res.function_num = 0xDE ;            /* 指示错误发生                */
    pMSPdu->ms_neg_res.err_decode = 0x80 ;              /* Error_Decode ,指明是 PROFIBUS */
                                                        /* DP/DPV1 类错误              */
    pMSPdu->ms_neg_res.err_code1 = DPV1_ERRCL_ACCESS | DPV1_ERRCL_ACC_INV_INDEX ;
                                                        /* Error_code1 ,支持是接入错，  */
                                                        /* 接入了非法的 SLOT ；          */
    pMSPdu->ms_neg_res.err_code2 = 0x00 ;               /* Error_code2 ,               */
}
break;
            /*
             *  请求读当前 16 位计数器上下限
             */
case 4:
if(preq->ms_wr_rd_tr.index == 1){
    memcpy(pMSPdu->ms_wr_rd_tr.pdu_data,&user.cont_lim_L,2);
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
        pMSPdu->ms_wr_rd_tr.length = 2 ;                    /* 必须写实际可能提供的读数据长 */
                                                            /* 度                          */
        pMail->Len += 2 ;
    }else if (preq->ms_wr_rd_tr.index == 2){
        memcpy(pMSPdu->ms_wr_rd_tr.pdu_data,&user.cont_lim_H,2);
        pMSPdu->ms_wr_rd_tr.length = 2 ;                    /* 必须写实际可能提供的读数据长 */
                                                            /* 度                          */
        pMail->Len += 2 ;
    } else {                                                /* 非法 index                  */
        pMSPdu->ms_neg_res.function_num = 0xDE ;            /* 指示错误发生                */
        pMSPdu->ms_neg_res.err_decode = 0x80 ;              /* Error_Decode ,指明是 PROFIBUS */
                                                            /* DP/DPV1 类错误              */
        pMSPdu->ms_neg_res.err_code1 = DPV1_ERRCL_ACCESS | DPV1_ERRCL_ACC_INV_INDEX ;
                                                            /* Error_code1 ,支持是接入错，  */
                                                            /* 接入了非法的 SLOT ；         */
        pMSPdu->ms_neg_res.err_code2 = 0x00 ;               /* Error_code2 ,               */
    }
    break;
            /*
             * 请求读从站地址
             */
    case 5:
    if(preq->ms_wr_rd_tr.index == 1){
                                                            /* memcpy(pMSPdu->ms_wr_rd_tr.p */
                                                            /* du_data,&user.cont_lim_L,2); */
        pMSPdu->ms_wr_rd_tr.pdu_data[0] = rd_address();
        pMSPdu->ms_wr_rd_tr.length = 1 ;                    /* 必须写实际可能提供的读数据长 */
                                                            /* 度                          */
        pMail->Len += 1 ;
    } else {                                                /* 非法 index                  */
        pMSPdu->ms_neg_res.function_num = 0xDE ;            /* 指示错误发生                */
        pMSPdu->ms_neg_res.err_decode = 0x80 ;              /* Error_Decode ,指明是 PROFIBUS */
                                                            /* DP/DPV1 类错误              */
        pMSPdu->ms_neg_res.err_code1 = DPV1_ERRCL_ACCESS | DPV1_ERRCL_ACC_INV_INDEX ;
                                                            /* Error_code1 ,支持是接入错，  */
                                                            /* 接入了非法的 SLOT ；         */
        pMSPdu->ms_neg_res.err_code2 = 0x00 ;               /* Error_code2 ,               */
    }
    break;
            /*
             * 不存在的 slot
             */
    default:
    pMSPdu->ms_neg_res.function_num = 0xDE ;                /* 指示错误发生                */
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
    pMSPdu->ms_neg_res.err_decode = 0x80 ;                    /* Error_Decode ,指明是 PROFIBUS */
                                                              /* DP/DPV1 类错误                */
    pMSPdu->ms_neg_res.err_code1 = DPV1_ERRCL_ACCESS | DPV1_ERRCL_ACC_INV_SLOT ;
                                                              /* Error_code1 ,支持是接入错，   */
                                                              /*  接入了非法的 SLOT ；         */
    pMSPdu->ms_neg_res.err_code2 = 0x00 ;                     /* Error_code2 ,                 */
    break;
}
pMail->Data[pMail->Len] = CRC_Crc8( pMail->Data, pMail->Len );
return       pMail ;
}
```

用户有诊断或报警，需要按照规定格式组织数据，然后调用 Alarm_Stat_Send（）函数来完成。以下提供 DEMO 函数来示例：

```
UBYTE user_demo_diagnostic( UBYTE diag_nr, UBYTE module_nr )
{
UBYTE alarm_appears[7] ;
UBYTE alarm_disappears[7] ;
UBYTE status_appears[3];
UBYTE status_disappears[3] ;
static UBYTE alarm_seq_nr;
static UBYTE status_seq_nr;
ALARM_STATUS_PDU alarm;
UBYTE Ind ;
UBYTE diag_len;
UBYTE ret_value;
UBYTE data i ,j;

ret_value = 1;
Ind = 0 ;
switch( diag_nr )
{
    case 1:
    {
                /*
                 * 有报警，有通道相关诊断数据。
                 */
        alarm_seq_nr = (alarm_seq_nr + 1) & (MAX_SEQ_NR - 1);
        alarm.header          = 11;
        alarm.type            = ALARM_TYPE_DIAGNOSTIC;
        alarm.slot            = module_nr;
        alarm.spec            = SPEC_APPEARS + (alarm_seq_nr << SPEC_SEQ_START);
        alarm.user_data_len   = 7;
        alarm.user_data_ptr   =     alarm_appears;
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
        alarm_appears[0] = user.cont_step ;                    /* 计数器当前时基                */
        memcpy(&alarm_appears[1],&user.counter,2);             /* 计数器值                      */
        memcpy(&alarm_appears[3],&user.cont_lim_L,2);          /* 计数器当前下限值              */
        memcpy(&alarm_appears[5],&user.cont_lim_H,2);          /* 计数器当前上限值              */

        diag_len =      0 ;
        j = 0 ;
        for(i=0;i<8;i++)                                       /* 模拟附带的对输入通道的诊断，  */
                                                               /* 如果拨码开关位为 0，假设其为  */
                                                               /* 断路                          */
        {
            if((readin()&(1<<i))){

            }else{
                user.diag_data[0+j*3]      =   0x82;           /* 通道相关的诊断头字节,模块 2 出 */
                                                               /* 问题                          */
                user.diag_data[1+j*3] =   0x40 + (i);          /* 故障通道，输入                */
                user.diag_data[2+j*3]      =   0x20 + 0x06;    /* 通道类型为位，故障类型为断路  */
                diag_len+=3 ;
                j++ ;
            }
        }

        Ind = ALARM_STATUS|DIAG_CHANNA ;
        break;
    }
    case 2:
    {

                /*
                 * 报警消失，
                 */
        alarm_seq_nr = (alarm_seq_nr + 1) & (MAX_SEQ_NR - 1);
        alarm.header            = 9;
        alarm.type              = ALARM_TYPE_DIAGNOSTIC;
        alarm.slot              = module_nr;
        alarm.spec              = SPEC_DISAPPEARS + (alarm_seq_nr << SPEC_SEQ_START)+
SPEC_ADD_ACK ;
        alarm.user_data_len   = 5;
        alarm.user_data_ptr   =      alarm_disappears ;

        alarm_disappears[0] = user.cont_step ;                 /* 计数器当前时基                */
        memcpy(&alarm_disappears[1],&user.cont_lim_L,2);       /* 计数器当前下限值              */
        memcpy(&alarm_disappears[3],&user.cont_lim_H,2);       /* 计数器当前上限值              */
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
        diag_len = 0 ;
        Ind =ALARM_STATUS;
        break;
}
case 3:
{
                /*
                 * 有诊断,出现地址开关被移动
                 */

        status_seq_nr = (status_seq_nr + 1) & (MAX_SEQ_NR - 1);
        alarm.header            = 5;
        alarm.type              = STATUS_MESSAGE;
        alarm.slot              = module_nr;
        alarm.spec              = SPEC_APPEARS + (status_seq_nr << SPEC_SEQ_START);
        alarm.user_data_len     = 1;
        alarm.user_data_ptr     = status_appears;

        status_appears[0]       = rd_address();

        diag_len = 0 ;

        Ind = ALARM_STATUS ;
        break;
}
case 4:
{
                /*
                 * 诊断消失,地址开关被被还原
                 */
        status_seq_nr = (status_seq_nr + 1) & (MAX_SEQ_NR - 1);
        alarm.header            = 5;
        alarm.type              = STATUS_MESSAGE;
        alarm.slot              = module_nr;
        alarm.spec              = SPEC_DISAPPEARS + (status_seq_nr << SPEC_SEQ_START);
        alarm.user_data_len     = 1;
        alarm.user_data_ptr     = status_disappears;
        status_disappears[0] = rd_address();
        diag_len = 0 ;
        Ind = ALARM_STATUS ;
        break;
}
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
    default:
    {
        break;
    }
}
    /*
     * 用户需要调用此函数来完成诊断数据的发送
     */
Alarm_Stat_Send(Ind ,&alarm,user.diag_data , diag_len);
return ret_value;
}
```

最后，用户需要在 main.c 中定义 main()函数，用户需要参照如下框架来设计您的主循环函数。

```
int main(void)
{
volatile unsigned char temp = 0;
UBYTE SEND = 0   ;
    /*
     * 准备准备设备组态数据 CFG
     */
   unsigned char cfg_data[] = {0x20,0x10,0x50};
    /*
     * 串口初始化及用户自己的初始化函数
     */
    UartInit();
    memset(&Uart,0,sizeof(STRUC_UART));
    /*
     * 调用模块初始化函数，需传入配置数据长度
     */
    PBIniti(cfg_data,3);
    /*
     * 如果用户设置了配置数据为 0，需要在此处读回来自主站的配置数据
     */
    //ReadeCFG();
    while (1)
    {
        /*-------------------------------------------------------------*
         * 用户在此添加循环通信输入数据                                *
         *-------------------------------------------------------------*/
        /*
         * 准备新的输入数据
         */
        sys.In_data[0] = readin();
        memcpy(&sys.In_data[1],&user.counter,2);
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
        /*----------------------------------------------------------------*
         * 用户在此添加诊断 ,发送诊断后，必须调用一次 RecvProcess()          *
         *----------------------------------------------------------------*/
        if (rd_address()!= sys.Device.Address)
        {
            /*
             * 用户动用了地址开关，报告状态.且只报告一次，不要重复报告同样的诊断
             */
            if (SEND == 0)
            {
                SEND ++ ;
                User_demo_diagnostic(3, 5);
                RecvProcess();
            }
        } else {
            if (SEND!=0)
            {
                SEND = 0 ;
                User_demo_diagnostic(4, 5);                    /* 报告状态消失
*/
                RecvProcess();
            }
        }
        /*----------------------------------------------------------------*
         * 用户调用该函数更新输入数据和非周期通信应答   *
         *----------------------------------------------------------------*/
    DataUpdate();
        /*----------------------------------------------------------------*
         * 用户调用该函数接收输出数据和非周期通信请求   *
         *----------------------------------------------------------------*/
        /*
         * 等待接收完成，一发送，必须得到一应答
         */
    RecvProcess();
        /*----------------------------------------------------------------*
         * 对输出数据的处理                                    *
         *----------------------------------------------------------------*/
    showout(sys.Out_data[0]);

}
return 1;
}
```

至此，用户更改几处必要的代码，就完成了对 DPM 模块通信的操作。现在，你只需要写好您自己的应用代码就成功完成了整个项目了。

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

4.4 修改GSD

当您开发完成您的产品后，您需要写设备的 GSD 描述文件，在本章，我们将通过实例来说明您如何修改 sample.GSD 文件到您自己的设备 GSD。

GSD文件是可以使用文本文件打开并编写的，但建议您使用GSDEdit 软件，在该软件中，有详细的关键字说明帮助。如图 4.2所示：

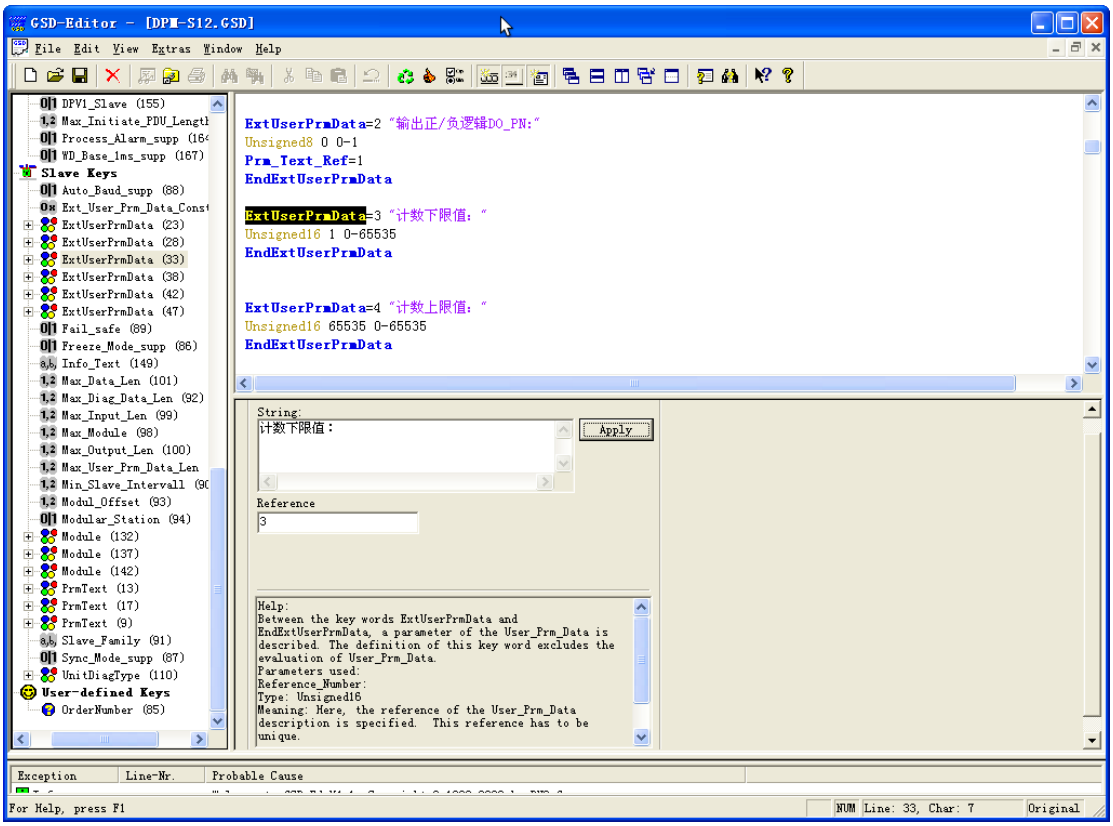

图 4.2 GSD 编辑器

```
;=================================================================
; GSD 文件 : XGate-DPS 致远电子有限责任公司
; 产品型号: XGate-DPS; 版本 : 2009 年 11 月 V1.1
; 文件名 : XGate-DPS.GSD
;=================================================================
#Profibus_DP
; User-Parameter-Definition ; 该部分是显示在组态软件中的，该部分由用户设计更改
PrmText=1
Text(0)="正逻辑(高有效)"
Text(1)="负逻辑(低有效)"
EndPrmText
PrmText=2
Text(0)="1ms"
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
Text(1)="10ms"
EndPrmText
PrmText=3
Text(1)="enable"
Text(0)="disable"
EndPrmText

ExtUserPrmData=1 "输入正/负逻辑 DI_PN:"
Unsigned8 0 0-1
Prm_Text_Ref=1
EndExtUserPrmData

ExtUserPrmData=2 "输出正/负逻辑 DO_PN:"
Unsigned8 0 0-1
Prm_Text_Ref=1
EndExtUserPrmData

ExtUserPrmData=3 "计数下限值："
Unsigned16 1 0-65535
EndExtUserPrmData

ExtUserPrmData=4 "计数上限值："
Unsigned16 65535 0-65535
EndExtUserPrmData

ExtUserPrmData=5 "计数基值:"
Unsigned8 0 0-1
Prm_Text_Ref=2
EndExtUserPrmData

ExtUserPrmData=6 " Alarm:"
Unsigned8 1 0-1
Prm_Text_Ref=3
EndExtUserPrmData

; Unit-Definition-List:
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
GSD_Revision=5
Vendor_Name="ZY    Ltd. Co." ;公司名，可按用户名修改
Model_Name="DPM-S1200I" ;模块名称，也是组态时该产品的名称，组态软件如 STEP 7，COM PROFIBUS
Revision="V1.0 "
Ident_Number=0x0F01          ;ID 号，必须与初始化报文一致
Protocol_Ident=0
Station_Type=0
Hardware_Release="H1.0 "    ;用户实际软硬件版本号
Software_Release="S1.0 "
9.6_supp=1
19.2_supp=1
93.75_supp=1
187.5_supp=1
500_supp=1
45.45_supp=1
1.5M_supp=1
3M_supp=1
6M_supp=1
12M_supp=1
MaxTsdr_9.6=60
MaxTsdr_19.2=60
MaxTsdr_45.45=250
MaxTsdr_93.75=60
MaxTsdr_187.5=60
MaxTsdr_500=100
MaxTsdr_1.5M=150
MaxTsdr_3M=250
MaxTsdr_6M=450
MaxTsdr_12M=800
Implementation_Type="VPC3+"
Bitmap_Device="SE_B"         ;图标文件，用户可以自制图标，缺省
; Slave-Specification:
OrderNumber="DPM-S1200I" ;产品序列号，可按用户名修改
Freeze_Mode_supp=1
Sync_Mode_supp=1
Auto_Baud_supp=1
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
Fail_safe=1
Min_Slave_Intervall=6
Slave_Family=9@TdF@DPM-S_Brige   ;组态中的分类名
Max_Diag_Data_Len=244
Modul_Offset=1
Modular_Station=1

;DP V1
Max_Module=6
Max_Input_Len=244
Max_Output_Len=244
Max_Data_Len=488
;User_Prm_Data_Len=13
Max_User_Prm_Data_Len=13
Ext_User_Prm_Data_Const(0)= 0x00,0x00,0x00
;===============================================================================
;==== Modulstatus ==============================================================
;===============================================================================
UnitDiagType=130
X_Unit_Diag_Area = 24 - 25
X_Value(1)= "Slot 1: Module error"
X_Value(2)= "Slot 1: Wrong module"
X_Value(3)= "Slot 1: No module"
X_Unit_Diag_Area_End
X_Unit_Diag_Area = 26 - 27
X_Value(1)= "Slot 2: Module error"
X_Value(2)= "Slot 2: Wrong module"
X_Value(3)= "Slot 2: No module"
X_Unit_Diag_Area_End
X_Unit_Diag_Area = 28 - 29
X_Value(1)= "Slot 3: Module error"
X_Value(2)= "Slot 3: Wrong module"
X_Value(3)= "Slot 3: No module"
X_Unit_Diag_Area_End
EndUnitDiagType
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
; Module-Definitions:  该部分有用户设计内容而定
Module= "DO8" 0x20;
1
Ext_Module_Prm_Data_Len=1
Ext_User_Prm_Data_Ref(0)=2
EndModule
Module="DI8" 0x10;
2
Ext_Module_Prm_Data_Len=1
Ext_User_Prm_Data_Ref(0)=1
EndModule
Module="Counter16" 0x50;
3
Ext_Module_Prm_Data_Len = 6
Ext_User_Prm_Data_Ref(0)=6
Ext_User_Prm_Data_Ref(1)=5
Ext_User_Prm_Data_Ref(2)=3
Ext_User_Prm_Data_Ref(4)=4
Info_Text="ZHIYUAN: counter module, Diag-alarm supported"
EndModule
;===============================================================================
;==== DPV1 KEY WORDS ===========================================================
;===============================================================================
DPV1_Slave          = 1
C1_Read_Write_supp     = 1
C1_Max_Data_Len              = 124          ;The parameter specifies the maximum length of user data
excluding
                                                ;Function_Num, Slot_number, Index, Length,
transferred on the
                                                ;MSAC_1 communication channel.
                                                ;Type: Unsigned8 (0 .. 240)
                                    ;用户不要修改
C1_Response_Timeout           = 300
Diagnostic_Alarm_supp         = 1
Process_Alarm_supp             = 1
Alarm_Type_Mode_supp          = 0
```


©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

```
Alarm_Sequence_Mode_Count    = 16
WD_Base_1ms_supp             = 1
Publisher_supp               = 1

C2_Read_Write_supp           = 1
C2_Max_Data_Len              = 124      ;The parameter specifies the maximum length of user data excluding
                                        ;Function_Num, Slot_number, Index, Length, transferred on the
                                        ;MSAC_2 communication channel.
                                        ;Type: Unsigned8 (0,48 .. 240)
                                        ;用户不要修改
C2_Response_Timeout          = 4000     ;用户按照当前使用波特率来改变：
                                                ;rate < 187.5 k          = 4000
                                                ;rate = 500 k            = 2000
                                                ;rate > 1.5M             = 1000
C2_Max_Count_Channels        = 3
Max_Initiate_PDU_Length      = 52       ;The parameter specifies the maximum length of an Initiate Request
                                        ;PDU including the Function_Num to the Resource Manager.
                                        ;Type: Unsigned8 (0,52.. 244)
DPV1_Data_Types              = 0

SlotDefinition
Slot(1) = "Slot 1" 1 1
Slot(2) = "Slot 2" 2 2
Slot(3) = "Slot 3" 3 3
EndSlotDefinition
```

至此，您编写好了您的设备描述文件，可以组网调试您的系统了。

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

5. 组态网络

5.1 主站平台

本例的主站平台：

硬件：CP5613A2 PCI 板卡，西门子；

软件：Step 7 、Smatic Net （CP5613A2 驱动）、DPMaster561X Demo 软件，西门子。

本例认为您已经安装好了这些硬件和软件，其具体步骤请参照相关说明。

5.2 组态系统

下面，以本开发平台为例详细如何组网调试系统。

1. 将 DPS-12.GSD 文件拷贝到“*:\STEP 7\S7DATA\GSD”；
2. 打开PC机的“开始”—“所有程序”—“Simatic”—“SIMATIC NET”—“Configuration Console”，在“Modules”下的“CP5613 A2”下，设置“Address”信息如图 5.1下：

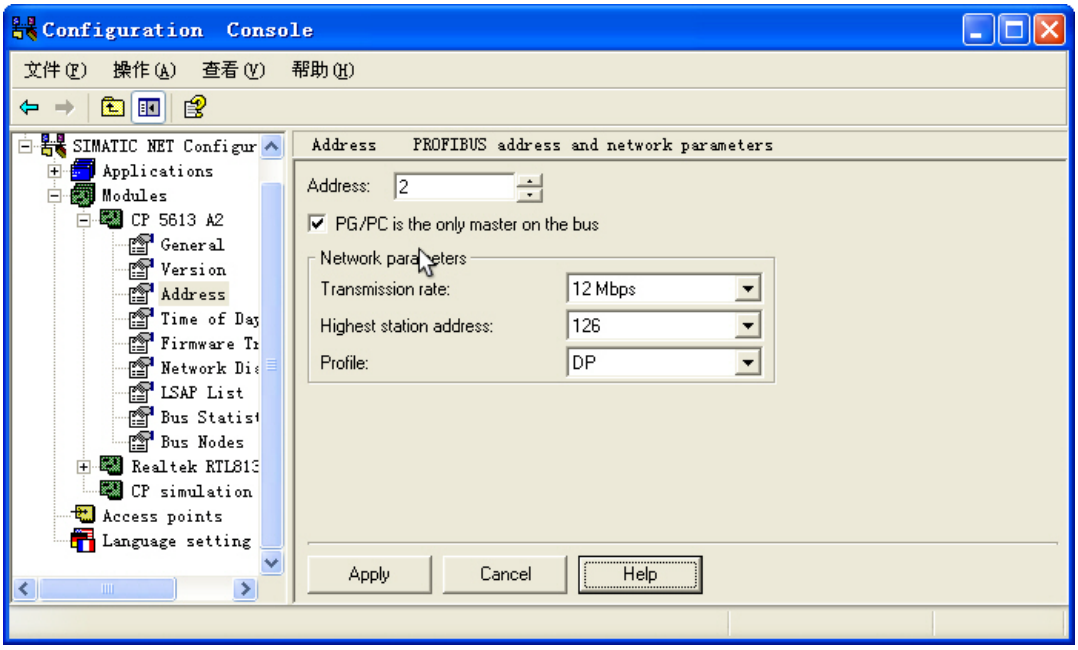

图 5.1 配置平台

3. 双击PC右下角“station configuration editor” ，弹出如图 5.2：

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

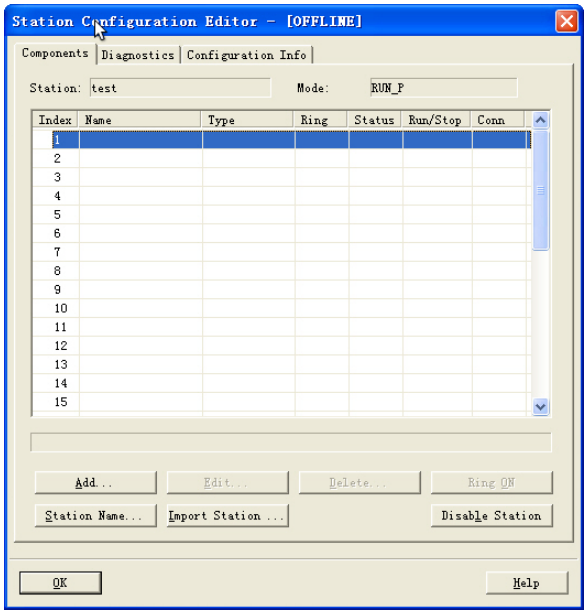

图 5.2 站编辑器

4. 点击“Station Name”,在弹出的对话框中输入“test”站名，点击“OK”，如图 5.3 所示；

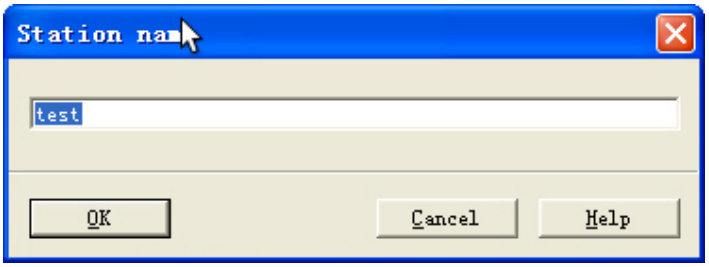

图 5.3 设置站名称

5. 在“Index 1”中点击鼠标右键，出现并点击“ADD”，选择如如图 5.4对话框：

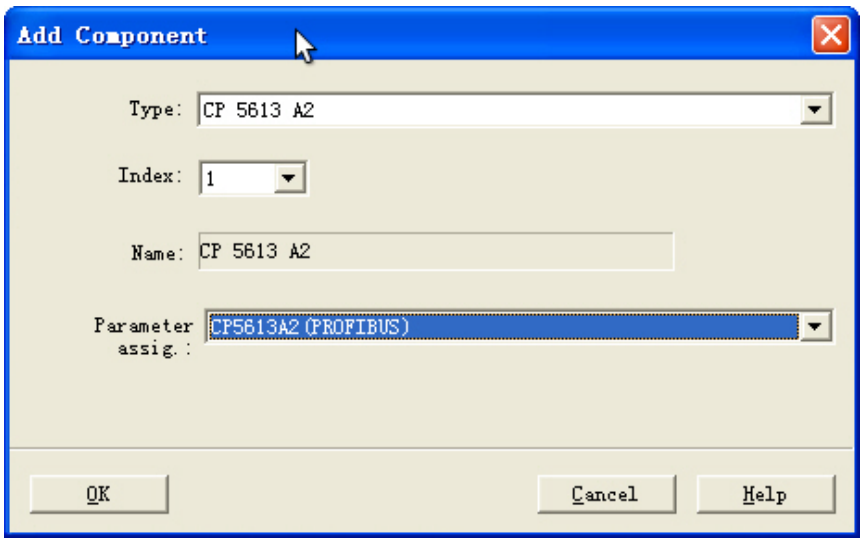

图 5.4 添加组件

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

6. 随后弹出的对话框设置如图 5.5：

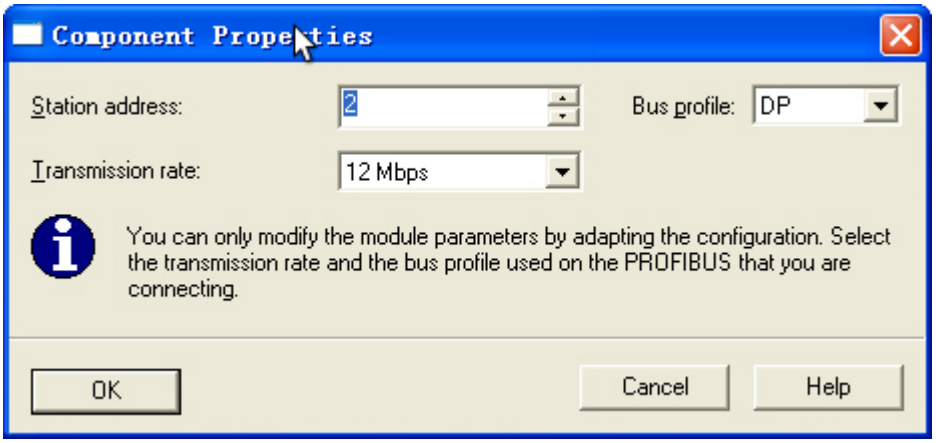

图 5.5 组件属性设置

7. 同样，在“Index 2”中添加“Application”，最终如下所示，单击“OK”，如图 5.6：

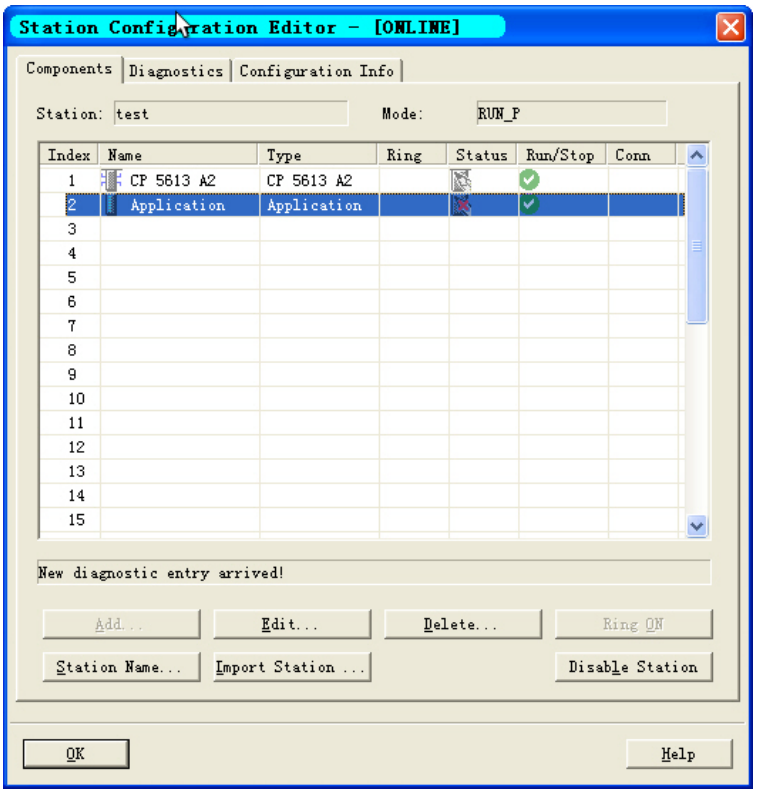

图 5.6 添加应用程序组件

8. 打开SIMATIC Manager，并选择“文件”—“新建项目”，填写“名称”和“存储位置”，然后确定，如图 5.7所示：

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

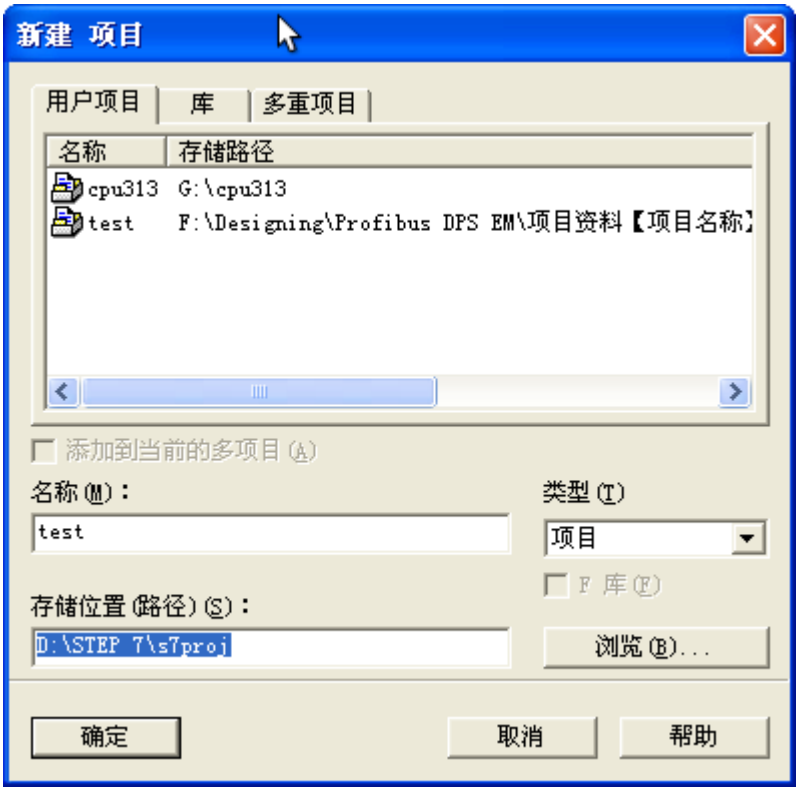

图 5.7 新建项目

9. 单击“test”选中，右键—“插入新对象”—“SIMATIC PC站点”，如图 5.8所示：

图 5.8 添加 PC 站点

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

10. 将站名命名为“test”，双击进入test站，然后双击“组态”，如图 5.9所示：

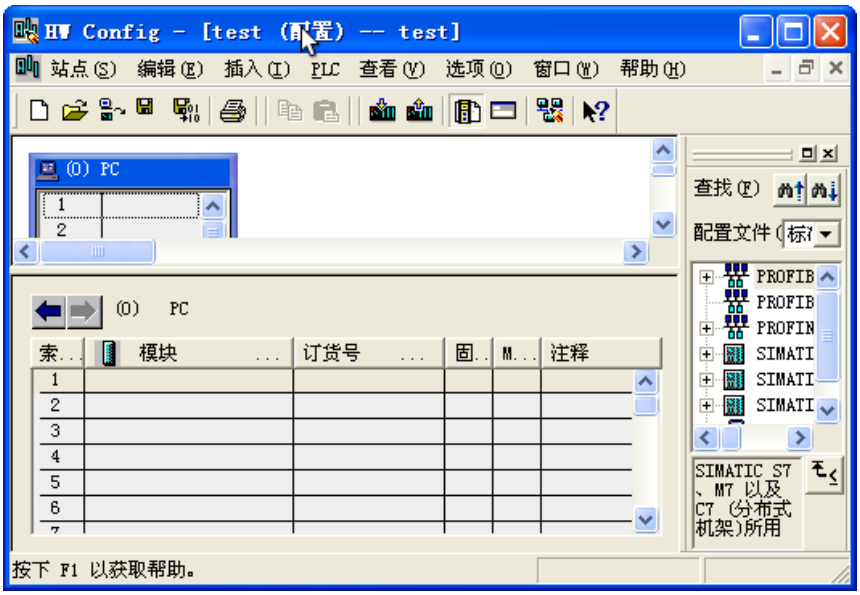

图 5.9 组态主界面

11. 在组态界面下，选择“选项”—“更新目录”来加载 GSD 文件；
12. 在右边“配置文件”下，从“SIMATIC PC Station”—“CP PROFIBUS”—“CP 5613 A2”下，将“SW V6.2”拖到“索引 1”中，如图 5.10所示；

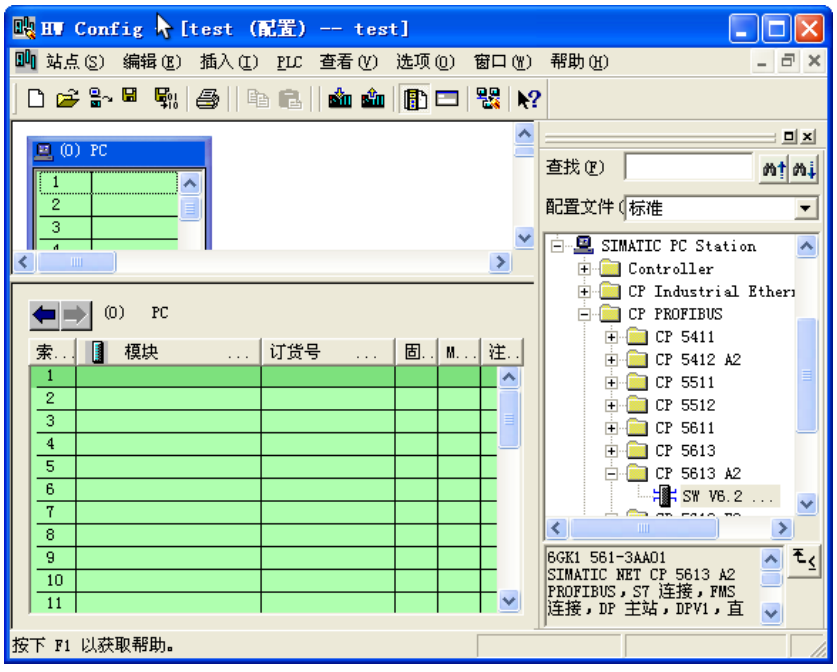

图 5.10 添加 PC 主站卡

13. 在弹出的对话框中，选择“Address”为 2，添加PROFIBUS 网络“PROFIBUS(1)”，然后确定，如图 5.11所示：

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

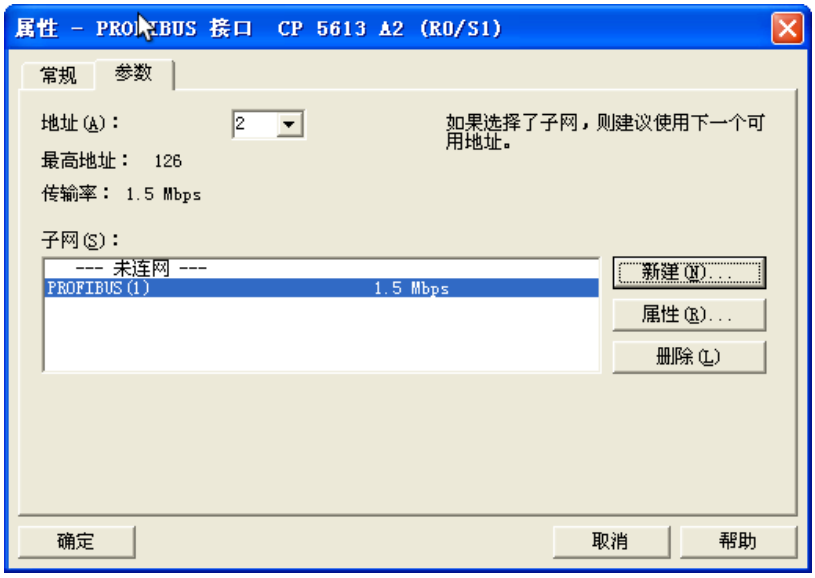

图 5.11 建立主站网络

14. 从“SIMATIC PC Station”—“User Application”—“Application”下，将“SW V6.3”拖到“索引 2”中；选中索引 1 的“CP5613A2”单击右键选择“添加主站系统”，选择“应用程序”，如图 5.12所示：

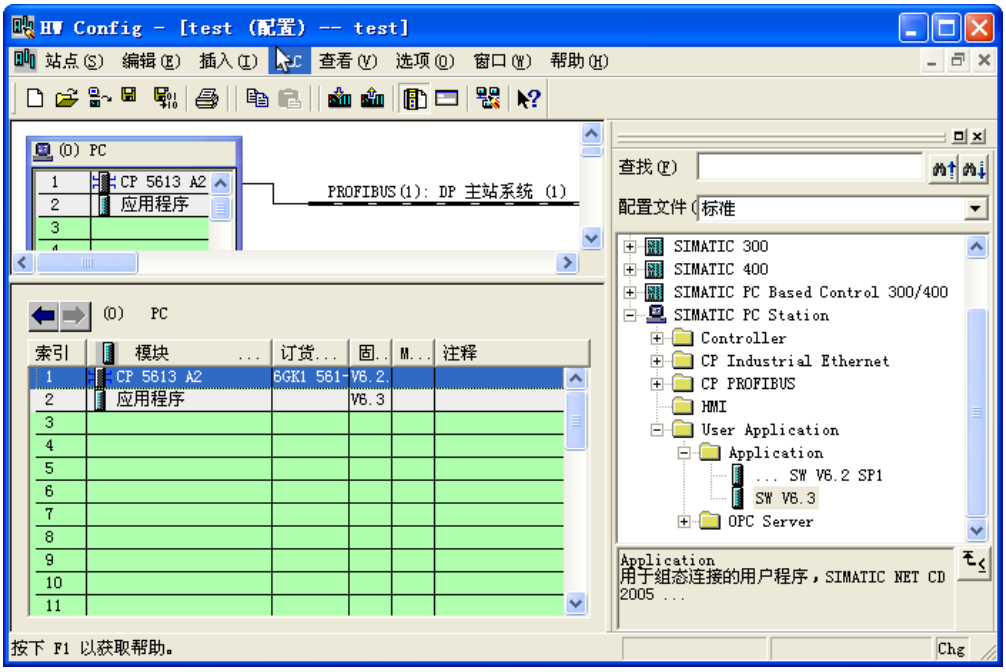

图 5.12 添加 PROFIBUS 应用

15. 从“PROFIBUS DP”—“Additional Field Devices”—“Geteway”—“DPM-S_Brige”下将“DPM-S1200I”拖到“PROFIBUS(1)”主站系统总线上，然后，设置站地址“6”，如图 5.13所示：

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

图 5.13 添加从站

16. 双击设备或模块，设置您的用户参数。

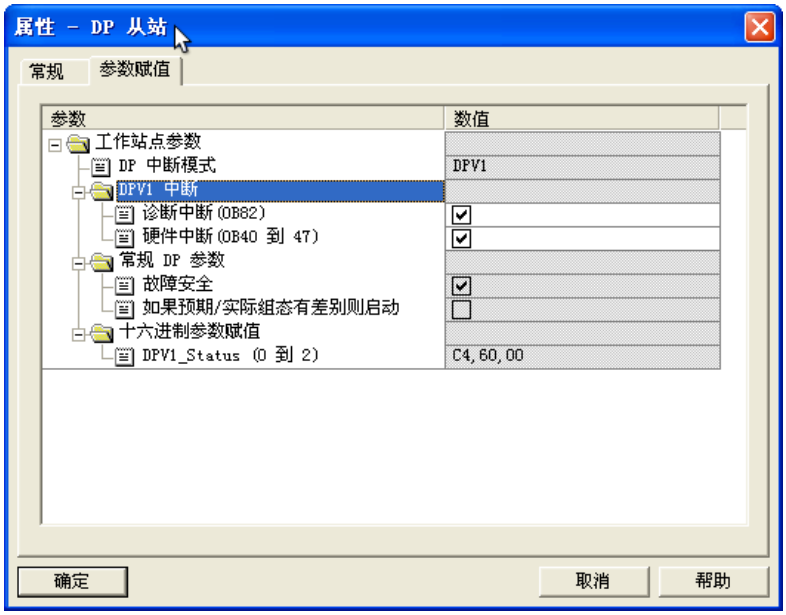

图 5.14 设置站参数

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

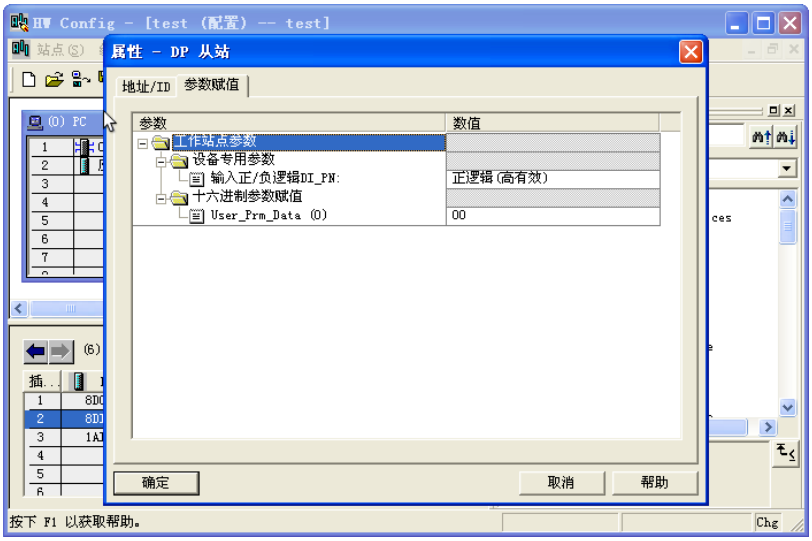

图 5.15 设置模块参数

17. 点击“站点”—“保存并编译”；再点击点击“PLC”—“下载” ，以后全选择“确定”即可，将配置下载到主站卡中：

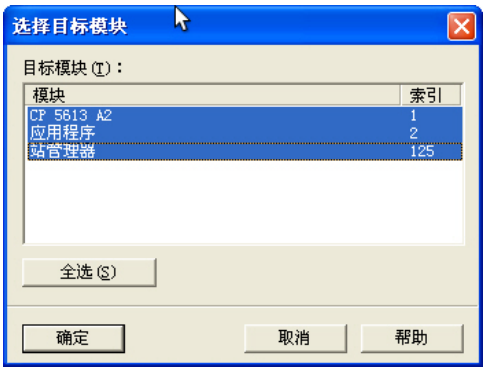

图 5.16 下载

18. 打开 PC 机软件 DPMaster561X.exe，然后在 DP Master 菜单下，选择 initial；

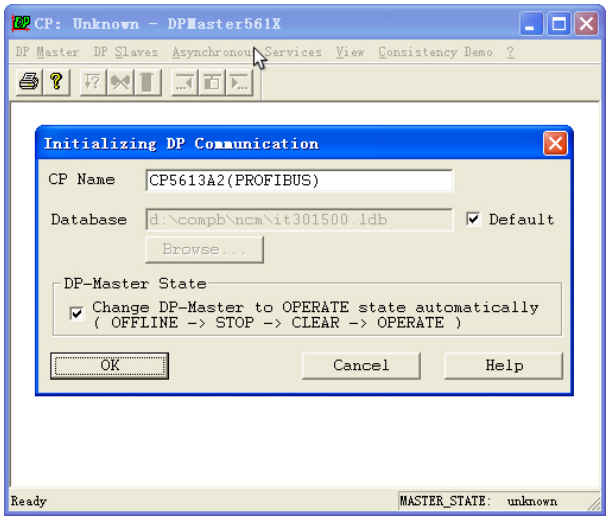

图 5.17 运行主站调试软件

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

19. 单击OK，接上EVB电源，当主站初始化完从站后，出现如下界面，此时，EVB上的PBF指示灯有红色变为绿色，DPM-S1200I模块上的DaEX指示灯亮表明建立了通讯，如图 5.18所示：

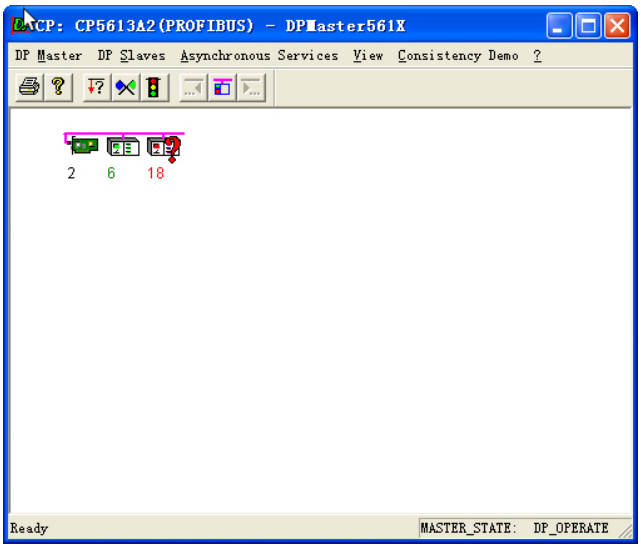

图 5.18 站 6 进入数据交换阶段

20. 双击绿色“6”号站：勾选如下图，先点击“Read”再点击“Write”，如图 5.19：

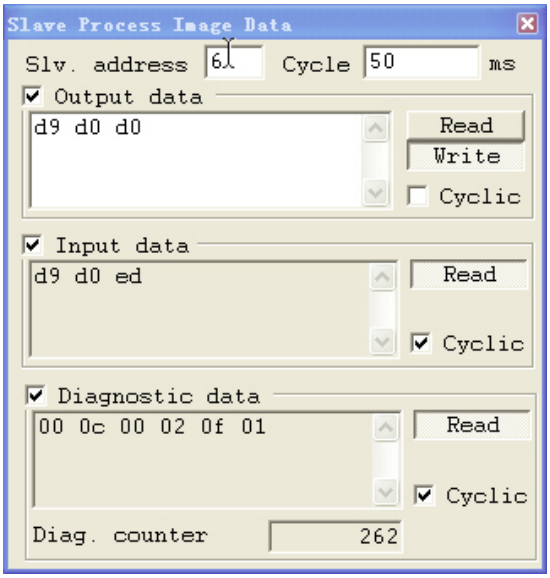

图 5.19 输入输出及诊断数据窗口

21. 勾选所有的“Cyclic”，“input data”的前两个自己会随着“output data”的变化而变化。拨动 EVB 上的 SW1 拨码开关，“input data”将显示相应的开关值;EVB 板上的 LED 灯组将输出当前接收到的“OUTPUT DATA”值。

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.

6. 免责声明

本文档提供有关致远电子产品的信息。本文档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除致远电子在其产品的销售条款和条件中声明的责任之外，致远电子概不承担任何其它责任。并且，致远电子对致远电子产品的销售和／或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。致远电子产品并非设计用于医疗、救生或维生等用途。致远电子可能随时对产品规格及产品描述做出修改，恕不另行通知。

该系列产品可能包含某些设计缺陷或错误，一经发现将收入勘误表，并因此可能导致产品与已出版的规格有所差异。如客户索取，可提供最新的勘误表。

在订购产品之前，请您与当地的致远电子销售处或分销商联系，以获取最新的规格说明。本文档中提及的含有订购号的文档以及其它致远电子文献可通过访问广州致远电子有限公司的万维网站点获得，网址是：

http://www.embedcontrol.com/ 或致电+86-20-22644249 查询。

©2008 Guangzhou ZHIYUAN Electronics CO., LTD.